C Piscine

Rush 02

Staff 42 pedago@42.fr

Summary: This document is the subject for Rush02 of the C Piscine @ 42.

Contents

1	Instructions	2
2	Foreword	3
3	Subject	4
4	Bonus	6
5	Super Bonus	7

Chapter 1

Instructions

- Each member of the group can register the whole group to the evaluation.
- The group MUST be registered to an evaluation.
- Be mindful of the submission procedures indicated at the start of every exercise.
- Moulinette compiles with these flags: **-Wall -Wextra -Werror**, and uses **gcc**.
- If your program doesn't compile, it will be graded 0.
- You must complete the project with the imposed team and show up at the evaluation slot you've selected, with all of your teammates.
- You project must be done by the time you get to the evaluation. The purpose of evaluation is for you to present and explain any and all details of your work.
- Each member of your group must be fully aware of the inner workings of the project. Should you choose to split the workload, make sure you all understand what everybody has done. During evaluation, you'll be asked questions, and the final grade will be based on the worst explanations.
- Gathering the group for work and evaluation is your responsibility.
- Moulinette relies on **norminette** to check if your files respect the Norm. An exercise containing files that do not respect the Norm will be graded 0.

For this rush, `norminette` is launched without any particular flag!

Chapter 2

Foreword

Here is a old-fashioned pecan pie recipe for you :

Ingredients
- Pastry dough
- 3/4 stick unsalted butter
- 1 1/4 cups packed light brown sugar
- 3/4 cup light corn syrup
- 2 teaspoon pure vanilla extract
- 1/2 teaspoon grated orange zest
- 1/4 teaspoon salt
- 3 large eggs
- 2 cups pecan halves (1/2 pound)
Accompaniment: whipped cream or vanilla ice cream

Preparation:
Preheat oven to 350°F with a baking sheet on middle rack.
Roll out dough on a lightly floured surface with a lightly floured rolling pin into a 12 inch round and fit into a 9 inch pie plate.
Trim edge, leaving a 1/2-inch overhang.
Fold overhang under and lightly press against rim of pie plate, then crimp decoratively.
Lightly prick bottom all over with a fork.
Chill until firm, at least 30 minutes (or freeze 10 minutes).
Meanwhile, melt butter in a small heavy saucepan over medium heat.
Add brown sugar, whisking until smooth.
Remove from heat and whisk in corn syrup, vanilla, zest, and salt.
Lightly beat eggs in a medium bowl, then whisk in corn syrup mixture.
Put pecans in pie shell and pour corn syrup mixture evenly over them.
Bake on hot baking sheet until filling is set, 50 minutes to 1 hour.
Cool completely.

Cooks notes:
Pie can be baked 1 day ahead and chilled. Bring to room temperature before serving.

Chapter 3

Subject

Turn-in directory : `ex00/`
Files to turn in: `Makefile and all necessary files`
Allowed functions: `write, read, malloc, free`

- Create a program that takes a character string as argument and detects which rush pattern it represents, as well as its dimensions. Rush patterns were defined in the Rush00 subject.
- Executable name: **`rush-2`**
- Your source code will be compiled as follows :

```
make fclean
make
```

- If the argument isn't a rush pattern, here's an example of output display :

```
$> echo "Is there a chance I'll get more than 0?" | ./rush-2
none
$>
```

- Whatever the answer, your line must be ended by a "\n"
- If the string matches more than one rush pattern, you must display them all alphabetically.

- Example :

```
$> ./rush-00 4 4
o--o
|  |
|  |
o--o
$> ./rush-00 4 4 | ./rush-2
[rush-00] [4] [4]
$> ./rush-01 3 4 | ./rush-2
[rush-01] [3] [4]
$> ./rush-02 1 1
A
$> ./rush-03 1 1
A
$> ./rush-04 1 1
A
$> ./rush-02 1 1 | ./rush-2
[rush-02] [1] [1] || [rush-03] [1] [1] || [rush-04] [1] [1]
$>
```

Chapter 4

Bonus

- Detection of any of those shapes is worth 2 points :

  1. rectangle
  2. square
  3. triangle
  4. lozenge

- This is on top of the mandatory subject.
- Example :

```
$> ./rush-02 3 3 | ./rush-2
[rush-02] [3] [3] || [square] [3] [3] || [rectangle] [3] [3]
$>
```

- Make sure you have enough test files in order to support your work. You must demonstrate how your program detects shapes, and that it is indeed your **work**.

Chapter 5

Super Bonus

This bonus doesn't require you having done the previous bonus, but the mandatory subject remains mandatory.

- Detection of any of those shapes is worth 2 points :

  1. sastantua
  2. reversed rectangle
  3. reversed square
  4. reversed triangle
  5. reversed lozenge
  6. reversed circle

- Make sure you have enough test files in order to support your work.
- A reversed shape is a shape made of spaces, the rest is made of characters.
- In this example, spaces are represented by dots ['.'] :

```
$> cat ./carre_inverse_5_5
.....
.zaO.
.zaO.
.zaO.
.....
$> cat ./carre_inverse_5_5 | ./rush-2
[inverse square] [5] [5] || [square] [3] [3]
```

- Any student who, on top of all these shapes, handles a reversed sastantua will get the ultimate grade of **42** for this project.